Software Usability

Course notes for CSI 5122 - University of Ottawa

2023-Deck I:
Usability and UX in the Software Engineering Process

Timothy C. Lethbridge
< Timothy.Lethbridge@uottawa.ca >
http://www.eecs.uottawa.ca/~tcl/csi5122

Lack of usability is the most critical problem facing software engineering

In other words, the biggest gains in

- software quality
- productivity
- cost-reduction
- user satisfaction
- profitability, etc.

… **would come from focusing on usability and related issues**.

Evidence for the assertion

Failures of projects due in significant part to usability

- E.g. FAA air traffic control systems

Observations of ‘great’ software vs. not-so-great with respect to usability

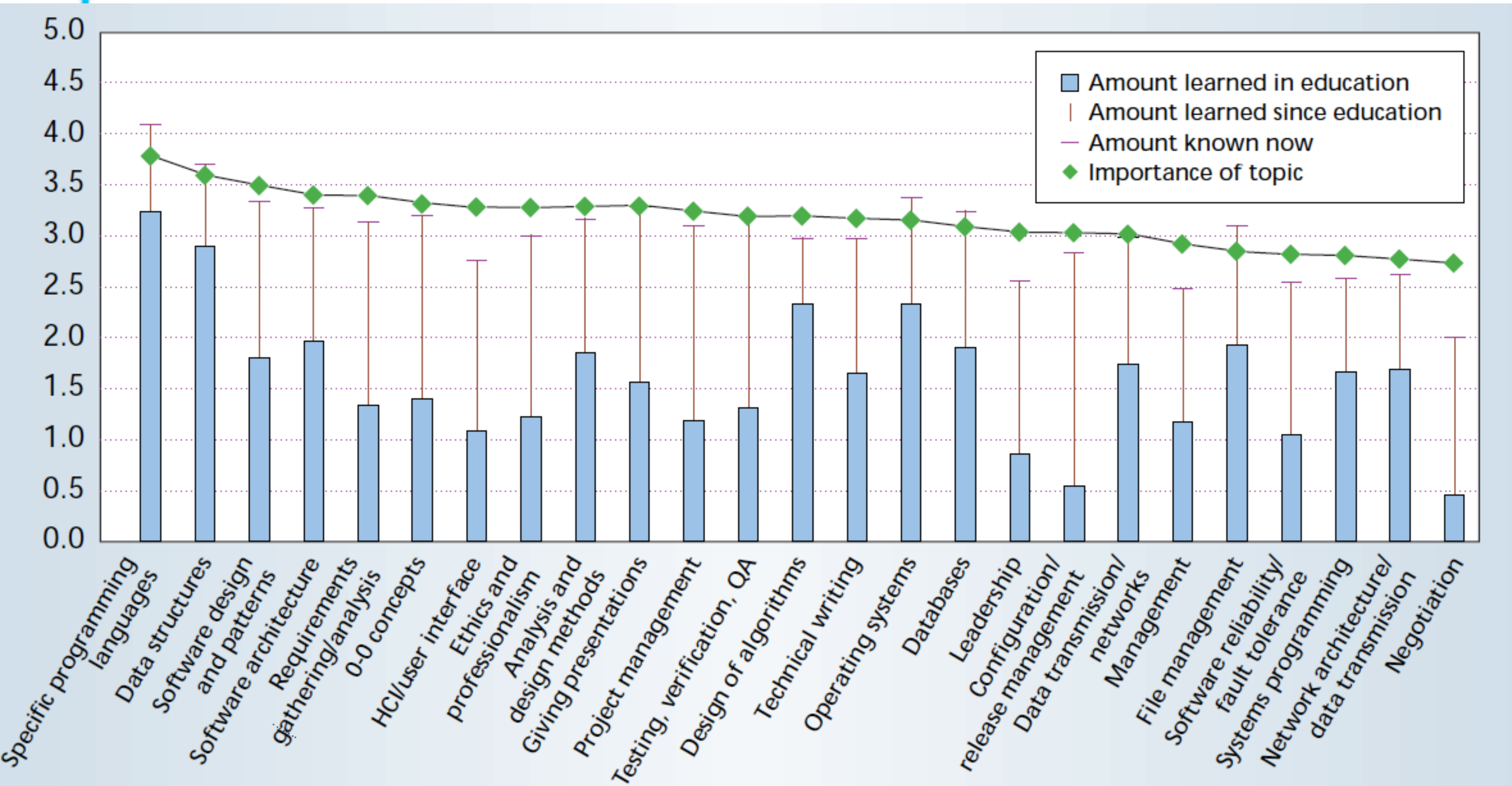
- Great:
  - —Google (many of its products, but some getting worse)
  - —Microsoft (some of it, e.g. Excel, Word, but many issues)
  - —Apple (much of it, but weaknesses creep in)
- Not-so-great:
  - —Much in-house custom software
  - —Software development environments (many aspects)
  - —Web sites (far too many of them)
- I have observed many users formally and informally

My survey of practitioners (next slide)

Analysis of benefits and cost savings

Practitioner survey about 75 topics taught to computer science students

Lethbridge, T.C. (2000), “What Knowledge is Important to a Software Engineer?”, *IEEE Computer*, May, pp. 44-50.



More on the survey

HCI was second in terms of "knowledge gap"

- Where importance most exceeds current knowledge

The top 5 (out of 75)

- Negotiation
- HCI/user interfaces
- Leadership
- Real-time system design
- Management

I believe not much has changed in the last two decades

Sources of resistance to adopting usability practices

Primarily: Inadequate education of students, practitioners and managers

Which leads to:

- Lack of introspection ability of software developers
  - —Unable to think like users
- Lack of integration of usability into core development processes
- Persistent beliefs
  - —Most software is usable enough
  - —User experience can be fixed at the “end”
  - —User experience should be left to the HCI experts

Causes of the problems in education

Faculty dont have the background either

The field of HCI is seen as distinct and separate

The field is perceived as too soft:

- There is a tendency to focus teaching and research on hard deterministic areas

There is insufficent industry push for research in this area, although this is changing

There is little employer pull for greater education

Some partial solutions to the education problem

The SE and HCI communities need to work more closely together

- Core UI development and evaluation topics should be considered jointly part of the two fields

Improve education by ensuring HCI permeates curriculum models, certifications, accreditation, etc.

Enable corporations to be certified for their usability capabilities

- Would help "pull" education up by creating a need for professionals
- Clients might come to learn that better software comes from such companies

Should user experience be considered a central skill for all software engineers? - 1

I argue yes:

- Usability should have no special status as compared to other qualities such as reliability, efficiency, maintainability, etc.
- Design involving users and their needs must *drive* software development

Should user experience be considered a central skill for all software engineers? - 2

Yes, but this does not preclude HCI specialists

- Analogies:
  - —A software engineer must be capable of designing architectural elements for storing data and information
    - But there will always need to be database specialists
  - —Similar argument can be made for
    - Security experts, performance experts, requirements experts, etc.

Core UX and UCD can be part of the SE field *and* part of a broader HCI field

RATING CORPORATE MATURITY WITH RESPECT TO USER EXPERIENCE

The UX Maturity Model (proposed by Lethbridge!)

A 5-level CMM-like categorization of organizations

—Based on their interaction with human end-users in software development

—(Each level beyond 1 builds on the previous levels)

- Level 1: Haphazard
- Level 2: Defined input from users / usability awareness
- Level 3: Iterative interaction with users / design for usability
- Level 4: Controlled and measured involvement of users
- Level 5: Continually improving usability

Level 1: Haphazard

If you can get users to use the system, then it is considered good enough!

Level 2: Defined input from users / usability awareness

Users involved **in requirements reviews**

- **Feedback from users** at reviews is incorporated into the next stage

Design team members have basic training **in usability and UX**

Design team adheres to usability standards for look and feel

Reuse **of well-understood controls, styles etc.**

Level 3: Iterative interaction with users / design for usability

Users actively **involved in decision making**

Use case / task / persona analysis

Competitive analysis

Design with careful attention to usability guidelines

Usability design decisions are carefully analyzed **and the** decisions are recorded

- E.g. options analysis, tables of pros and cons

Level 3: Continued

User input and feedback with repeated **prototypes**

- From paper prototypes to functional prototypes

Informal qualitative **observations of users or heuristic evaluations**

Discount usability engineering

Level 4: Controlled and measured involvement of users

User input and feedback at all stages

Design team has trained UX experts and/or cognitive scientists

Careful cost-benefit optimization including usability

Setting of quantitative usability objectives

- For all parts of the system that will be subject to regular or critical-situation use
- For learnability, efficiency of use, etc.

Level 4: Continued

UX design **with formal usability studies**

Measurement of usability to determine progress towards goals

Level 5: Continually improving usability

Active development of new UX and UI understanding and innovations

Formal experiments **to validate new UI modalities**

- metaphors, controls, widgets, styles etc.

Anthropological studies **of human tasks**

- to enable optimized UI design

Scientific study **of**

- Users and their work practices
- Usability and the software engineering process

Potential benefits of UX Maturity Model

It provides a visible *framework* that companies can use to incrementally improve their practices

If even a few companies adopted it and were certified, they might have a competitive advantage in some situations